

Le coin du développeur

Utilisation en ligne de commande

Avec un terminal (CLI)



En travaux ..



Dans un terminal Windows, tapez : 'habby.exe' ou 'python.exe habby.py' suivis de la commande souhaitée.

- LIST_COMMAND : liste toutes commandes disponibles ;
- CREATE_PROJECT : création d'un projet ;
- CREATE_HYD : création d'un fichier hydraulique .hyd ;
- CREATE_SUB : création d'un fichier substrat .sub ;
- MERGE_GRID_SUB : création d'un fichier habitat .hab ;
- RUN_HABITAT : calcul des valeurs d'habitat à partir d'un fichier .hab ;
- RUN_HS : calcul l'hydrosignature d'un fichier .hyd ou .hab ;
- EXPORT : lancer les exports à partir d'un fichier .hyd ou .hab ;
- RUN_ESTIMHAB : lance le calcul Estimhab ;
- RUN_STATHAB : lance le calcul Stathab ;
- RUN_FSTRESS : lance le calcul FStress.



Lien vers la doc sphinx

Quelques exemples :

```
CREATE_PROJECT
path_prj=C:\Users\user.name\Documents\HABBY_projects\DefaultProj_CLI
```

```
CREATE_HYD model=TELEMAC inputfile="input_file.txt" cut=True
outputfilename=cli_test.hyd
path_prj=C:\Users\user.name\Documents\HABBY_projects\DefaultProj_CLI
CREATE_HYD model=TELEMAC inputfile="input_file.slf" unit_list=5400.0,7200.0
cut=True outputfilename=cli_test.hyd outputfilename=input_file_spe.hyd
path_prj=C:\Users\user.name\Document\HABBY_projects\DefaultProj_CLI
```

```
CREATE_SUB substrate_mapping_method=polygon
inputfile=sub_durance_PolygonCemagrefPercent_gpkg\sub_durance_PolygonCemagrefPercent_gpkg.gpkg
path_prj=C:\Users\user.name\Document\HABBY_projects\DefaultProj_CLI
```

```
MERGE_GRID_SUB hyd=input_file_spe.hyd
sub=sub_durance_PolygonCemagrefPercent_gpkg\sub_durance_PolygonCemagrefPercent_gpkg.sub
```



```
outputfilename=a1_a5_a9_sub_durance_PolygonCemagrefPercent_gpkg.hab
path_prj=C:\Users\user.name\Document\HABBY_projects\DefaultProj_CLI
```

```
RUN_HABITAT hab=a1_a5_a9_sub_durance_PolygonCemagrefPercent_gpkg.hab
pref_file_list=biology\models\ABL01.xml,biology\models\ABL01.xml,biology\models\ABL01.xml,biology\models\BAM01b.xml,biology\models\BAM01b.xml,biology\models\BAM01b.xml,biology\models\BAM01.xml,biology\models\BAM01.xml,biology\models\BAM01.xml
stage_list=adult,fry,juvenile,[0,45[,[120,270[,[45,120[,adult,fry,juvenile
hyd_opt=HV,HV,HV,HV,HV,HV,HV,HV,HV
sub_opt=Dominant,Dominant,Dominant,Coarser,Coarser,Coarser,Coarser,Coarser,C
oarser path_prj=C:\Users\user.name\Document\HABBY_projects\DefaultProj_CLI
```

```
RUN_ESTIMHAB 0.09 1.5 9.26 12.42 0.23 0.37 1.5 0.04 1.2,2.6,0.1
path_prj="C:\Users\user.name\Documents\HABBY_projects\DefaultProj_CLI"
RUN_ESTIMHAB 0.09 1.5 9.26 12.42 0.23 0.37 1.5 0.04
"...discharge_chronicle_date.txt"
path_prj="C:\Users\user.name\Documents\HABBY_projects\DefaultProj_CLI"
```

2021/03/01 16:37 · qroyer

Avec Python

 En travaux .. 

Lien vers docstring Sphinx

2021/03/01 16:37 · qroyer
2021/03/01 16:39 · qroyer

Collaboration au développement d'HABBY

Participer au développement Python d'HABBY

Préambule

Vous pouvez soumettre des demandes de modifications du code Python sur le projet HABBY <https://github.com/Yannlrstea/habby>:

- Si vous souhaitez de nouvelles fonctionnalités.
- Si vous avez trouvé un bug et l'avez corrigé vous-même.

2021/03/12 08:48 · qroyer

Création d'un environnement Python

Dépendances

Le projet Python HABBY est hébergé sur Github accessible au lien suivant : <https://github.com/Yannlrstea/habby>.

- Python >= 3
- Git

Aide à la création de l'environnement Python pour Windows

- Télécharger la wheel GDAL : <https://github.com/cgohlke/geospatial-wheels>
 - <https://www.gisinternals.com/release.php>
- Installer Microsoft Visual C++ 14.0 : <https://visualstudio.microsoft.com/fr/visual-cpp-build-tools/> si besoin
- Editez le fichier C:\habby_dev\habby\tips\virtual_envs\creation_env_habby_pip_py311.bat' et spécifiez :
 - le chemin d'accès à votre la wheel GDAL que vous avez préalablement téléchargée en remplaçant le chemin de la variable : 'gdal_wheel_path=' ;
 - sauvegardez le fichier.
- Lancer le fichier ' C:\habby_dev\habby\tips\virtual_envs\creation_env_habby_pip_py311.bat'.
- Si toutes les étapes se sont bien déroulées, vous devriez voir apparaître la fenêtre principale d'HABBY.
- Votre environnement virtuel Python pour HABBY est prêt.

Aide à la création de l'environnement Python pour Linux

- installation de GDAL for Linux and GDAL for Python (<https://mothergeo-py.readthedocs.io/en/latest/development/how-to/gdal-ubuntu-pkg.html>)

```
sudo apt-get install python3.6-dev
sudo add-apt-repository ppa:ubuntugis/ppa
sudo apt-get update
sudo apt-get install gdal-bin
ogrinfo --version
sudo apt-get install libgdal-dev
```

- Lancer le fichier 'tips\virtual_envs\creation_env_habby_pip.sh'.

```
pip3 install pip --upgrade
pip3 install virtualenv
cd ../habby_dev
virtualenv --python /usr/bin/python3.6 env_virtuels/env_habby_dev_pip
source env_virtuels/env_habby_dev_pip/bin/activate
pip3 install gdal==X.X.X --global-option=build_ext --global-option="-I/usr/include/gdal/"
pip3 install -r habby/requirements.txt
python habby/habby.py
```

2021/03/11 21:55 · qroyer

Structure du projet Python

Préambule

Le projet Python HABBY contient les fichiers suivants :



2021/04/22 09:51 · qroyer

Astuces

Réactivation des 'print'

Dans certaines fonctions lancées en multiprocessing la fonction 'print' est modifiée pour renvoyer les warnings et erreurs au processus principale.

Pour la remettre par défaut cette fonction 'print' lors d'un débogage par exemple, utilisez le code suivant :

```
sys.stdout = sys.__stdout__ # import sys
```

Réactivation du débogage pour les sous-process en QThread

```
RecursionError: maximum recursion depth exceeded while calling a Python object
```

Si vous apercevez cette erreur, c'est que vous souhaitez déboguer dans un sous-process en QThread.

Pour palier à cette erreur il faut renommer temporairement le nom de la méthode "run" de la classe "MyProcessManager" du fichier "src/process_manager_mod.py" par le nom "start".

2021/04/22 09:14 · qroyer

Utilisation de GIT

- Configuration nom ou ID Github :

```
git config --global user.name "John Doe"
```

- Configuration adresse mail ou adresse mail du compte Github :

```
git config --global user.email johndoe@example.com
```

- Clonage du projet Github en local : Pour Windows dans un répertoire créé C:\habby_dev :

```
git clone https://github.com/YannIrstea/habby
```

- Clonage du projet Github en local avec autorisation de modification sur Github : Pour Windows dans un répertoire créé C:\habby_dev :

```
git clone https://<TOKEN KEY>@github.com/YannIrstea/habby.git
```

- Unordered List ItemQuelques exemples:

```
git status : connaitre l'état actuel du projet
git checkout dev1 : changer de branche vers la branche 'dev1' du projet
git checkout master : changer de branche pour revenir à la branche master du projet
git pull : mettre à jour le projet local avec les dernières modifications disponible sur Github
git gui : ouvrir l'interface graphique de git pour réaliser les 'commits'
git push : mettre à jour le projet sur Github avec les dernières modifications du projet en local (nécessite une autorisation avec une clef TOKEN)
```

2021/11/17 11:34 · qroyer

Traduction du logiciel

Prés-requis

- Linguist.exe : (<https://github.com/thurask/Qt-Linguist/releases>)
- environnement Python HABBY avec PyQt5

Utilisation dans le code

Traduire des champs dans des classes Qt

```
self.tr('string to translate')
```

Traduire des champs en dehors des classes Qt

Si pas de classe ou héritage de classe qui pose problème :

```
from PyQt5.QtCore import QApplication
text = QApplication.translate('Input', 'string to translate') # 'Input'
sera le nom de la 'fausse' classe dans QLinguist et 'Neglect' le string à traduire.
```

Ou

```
app = QApplication(sys.argv)
```

```
languageTranslator = QTranslator(app)
if language == 0:
    input_file_translation = 'Zen_EN'
    languageTranslator.load(input_file_translation,
os.path.join(os.getcwd(), 'translation'))
if language == 1:
    input_file_translation = 'Zen_FR'
    languageTranslator.load(input_file_translation,
os.path.join(os.getcwd(), 'translation'))
elif language == 2:
    input_file_translation = 'Zen_ES'
    languageTranslator.load(input_file_translation,
os.path.join(os.getcwd(), 'translation'))
app.installTranslator(languageTranslator)
app.translate('Input', 'string to translate')
```

Mise à jour des fichiers .ts à traduire dans HABBY

- Vérifier que le fichier 'habby_trans.pro' contient bien les fichiers.py contenant les champs à traduire
- Pour mettre à jour les fichiers, lancer dans l'environnement virtuel et dans le répertoire 'habby', lancer la commande :

```
python -m PyQt5.pylupdate_main habby_trans.pro
```

- Ouvrir le fichier de langue souhaité (ex : 'Zen_FR.ts') dans le répertoire 'translation' avec Linguist.exe
- Dans le logiciel Linguist, renseigner les champs 'French translation' souhaités ;
- Sauvegarder le fichier ;
- Lancer l'invite de commande dans 'habby'
- Mettre à jour le fichier de langage choisi en lançant la commande (ici pour le fichier français) :

```
C:\habby_dev\dependance\linguist_5.13.2\lrelease.exe translation/Zen_FR.ts
```

```
* le fichier Zen_FR.qm utilisé par Qt a été mis à jour
* Relancer HABBY
```

2021/03/11 21:55 · qroyer

Création d'un exécutable HABBY

Windows

Pre-requis

- Librairie pyinstaller installée dans l'environnement Python
- Installer Inno Script Studio à partir de ce lien <https://www.kymoto.org/products/inno-script-studio/downloads>
- Installer Inno Setup à partir de ce lien <https://jrsoftware.org/isdl.php>

Windows

```
C:\habby_dev\habby\tips\executables\habby_to_exe_to_setup_pyinstaller.cmd
```

Linux

```
C:\habby_dev\habby\tips\executables\habby_to_exe_to_setup_pyinstaller.sh
```

Mac



2021/03/30 17:01 · qroyer

Changer et publier la nouvelle version d'HABBY

Numérotation et fonctionnement

La numérotation de la version du logiciel est de type versionnage sémantique X.Y.Z (ex: HABBY v1.1.1)

- X : Changements non rétrocompatibles. Obligatoire de re-cr ation des anciens projets HABBY.
- Y : Ajouts de fonctionnalit s r trocompatibles. Proposition de re-cr ation des anciens projets HABBY.
- Z : Corrections d'anomalies r trocompatibles.

 tape par  tape

Par exemple HABBY v1.1.1 ⇒ HABBY v1.2.0 :

1. fichier 'habby.py', ligne 27, changer la valeur X.Y.Z de la variable 'HABBY_VERSION_STR' ('1.1.1' ⇒ '1.2.0')
2. fichier 'tips\executables\setup_from_pyinstaller.iss', ligne 5, changer la valeur X.Y.Z de la variable 'MyAppVersion' ('1.1.1' ⇒ '1.2.0')
3. Commiter et pusher les modifications de code
4. Compiler les ex cutables et installeurs des diff rents syst mes d'exploitation
5. Uploader les ex cutables et changer le num ro de version X.Y.Z sur la page de t l chargement du wiki ('1.1.1' ⇒ '1.2.0')
6. Ajouter un tag de release sur '<https://github.com/Yannlrstea/habby/tags>' ('1.1.1' ⇒ '1.2.0')

2021/09/29 11:00 · qroyer

2021/03/01 16:44 · qroyer

Participer à la documentation d'HABBY

Technologie DokuWiki

Ce site internet est réalisé avec **DokuWiki** : <https://www.dokuwiki.org>

La syntaxe DokuWiki est la suivante : <https://www.dokuwiki.org/fr:wiki:syntax>

Syntaxe Dokuwiki HABBY

Pour se référer aux instructions concernant l'interface d'HABBY, la syntaxe est la suivante :

1. `<hi #47B5E6>Menu - Sous-menu</hi>` `<hi #47B5E6>**Menu - Sous-menu**</hi>`
2. `<hi #9BFFFF>Affichage interface</hi>` `<hi #9BFFFF>**Affichage interface**</hi>`
3. `'Valeur renseignée' //\'Valeur renseignée\'//`
4. `<hi #9BFFFF>[Bouton]</hi>` `<hi #9BFFFF>**[Bouton]**</hi>`
5. `<hi #47B5E6>[Bouton principal]</hi>` `<hi #47B5E6>**[Bouton principal]**</hi>`
6.  `En travaux ..`  `FIXME En travaux ..` `FIXME`

Création de page

- Créer un nouveau lien dans la table des matières de la page <https://habby.wiki.inrae.fr/fr:start> (ou <https://habby.wiki.inrae.fr/en:start> pour la version anglaise)
- Créer cette nouvelle page en cliquant sur le nouveau lien de la table des matières (surligné en rouge quand la page n'existe pas)
- Si la page créée se trouve dans un répertoire (de plusieurs pages) et pour conserver un ordre d'affichage, ajouter à la fin, avec N le numéro de la page (N), par ex '0' :

```
{{indexmenu_n>N}}
```

- Pour créer une page (représentant un répertoire) qui affiche le contenu de toute les pages du répertoire dans le bon ordre, créer la puis ajouter avec path, le chemin de la page, par ex 'fr:guide_utilisateur' :

```
{{namespace>path&order=indexmenu}}
```

Afficher une page existant sur la page actuelle

```
{{page>fr:manuel_reference:modeles_2d:sub_description:sub_carto&noheader}}
```

2021/03/01 16:50 · qroyer

2021/03/01 16:40 · qroyer

Fonctionnement d'HABBY

Calcul d'habitat à partir de modèle 2D

Les modèles hydrauliques

Introduction



2021/04/27 15:11 · qroyer

Description du fichier indexHYDRAU.txt



2021/04/27 15:11 · qroyer

Logiciels de modélisation hydraulique

TELEMAC

Pour rappel :

| | |
|---|--|
| Site web | www.opentelemac.org |
| Extension(s) de fichier | .res, .slf, .srf |
| Nombre de dimension | 2 |
| Nombre de tronçon | n (Un fichier TELEMAC ne comporte qu'un seul tronçon) |
| Unité(s) de temps | s (secondes) |
| Équation | BSV (Barré de Saint-Venant) |
| Méthode d'analyse numérique | Éléments finis |
| Maillage variable | Non (les coordonnées XY des nœuds ne varient pas en fonction des pas de temps (maillage fixe)) |
| Variation de la donnée altimétrique z possible | Oui |
| Variable(s) aux mailles | - |
| Autre(s) variable(s) exploitables(s) aux mailles | - |
| Variable(s) aux nœuds | z, h, v |
| Autre(s) variable(s) exploitables(s) aux nœuds | v_x , v_y , température, vitesse de friction |
| Nombre de point par maille | 3 |
| Sens de rotation des mailles | horaire |
| Condition(s) de simulation | permanent, transitoire |

2021/02/22 12:30 · qroyer

Détection des variables à partir des noms :

| HABBY | variables aux mailles | variables aux nœuds |
|--------------|------------------------------|-------------------------------|
| z | - | BOTTOM, FOND |
| h | - | WATER DEPT, HAUTEUR D'EAU |
| v | - | MEAN VELOCITY, VITESSE MOY |
| v_x | - | VELOCITY U, VITESSE U |
| v_y | - | VELOCITY V, VITESSE V |
| v_frict | - | VITESSE DE FROT, FRICTION VEL |
| temp | - | TEMP |

2021/04/27 15:09 · qroyer

HEC-RAS 2D

2021/04/27 15:13 · qroyer

Rubar2D

2021/04/27 15:13 · qroyer

Basement

Pour rappel :

| | |
|---|------------------|
| Site web | basement.ethz.ch |
| Extension(s) de fichier | .h5 |
| Nombre de dimension | 2 |
| Nombre de tronçon | |
| Unité(s) de temps | |
| Équation | |
| Méthode d'analyse numérique | Volumes finis |
| Maillage variable | |
| Variation de la donnée altimétrique z possible | Oui |
| Variable(s) aux mailles | z, h, v |
| Autre(s) variable(s) exploitables(s) aux mailles | |
| Variable(s) aux nœuds | |
| Autre(s) variable(s) exploitables(s) aux nœuds | |
| Nombre de point par maille | |
| Sens de rotation des mailles | |
| Condition(s) de simulation | |

2021/02/23 08:19 · qroyer

| HABBY | variables aux mailles | variables aux nœuds |
|---------|-----------------------|---------------------|
| z | BottomEl | Coordnts |
| h | - | - |
| v | - | - |
| v_x | - | - |
| v_y | - | - |
| v_frict | - | - |
| temp | - | - |

2021/04/27 15:09 · qroyer

2021/04/27 15:12 · qroyer

2021/04/27 15:10 · qroyer

2021/04/27 15:08 · qroyer

Calcul d'habitat à partir de modèles statistiques

A partir d'Estimhab



2021/04/27 15:15 · qroyer

A partir de Stathab



2021/04/27 15:15 · qroyer

A partir de FStress



2021/04/27 15:15 · qroyer

2021/04/27 15:14 · qroyer

Lecture d'un modèle hydraulique 1D ou 2D dans HABBY

HABBY va contrôler l'intégrité des données d'entrées hydrauliques et les modifier dans certains cas :

1. Lecture des fichiers de simulation hydraulique 1D ou 2D.
2. (Seulement pour les modèles 1D) Création d'un maillage triangulaire à partir des profils en travers.
3. (Seulement pour les modèles 2D et maillage non triangulaire) Triangulation pour avoir un maillage triangulaire.
4. (Seulement pour les modèles 2D et Volumes Finis) Conversion d'un maillage volume finis vers

un maillage élément finis.

5. Assemblage des tronçons et unités si plusieurs fichiers d'entrée.
6. Sauvegarde du maillage complet (whole_profile).
7. Détection d'un maillage variable en coordonnées XY entre chaque unité d'un même tronçon.
8. Détection d'un maillage variable en élévation Z entre chaque unité d'un même tronçon.
9. Pour chaque unité de chaque tronçon:
 1. Détection de doublons de mailles et nœuds à la lecture. Si oui, l'unité n'est pas acceptée par HABBY.
 2. Toute les valeurs de hauteurs d'eau inférieures au paramètre du projet spécifié par l'utilisateur, sont mise à 0. Toute les variables qui dépendent de la hauteur d'eau sont aussi mise à 0.
 3. Suppression des mailles sèches (hauteurs d'eau égales à 0.).
 4. (Seulement pour les modèles 2D avec l'option de projet activée) : Découpage des mailles semi-mouillées + détection de doublons de mailles et nœuds par sécurité.
 5. Calcul de l'aire de chaque maille.
 6. Suppression des mailles dont l'aire calculée est nulle.
10. Création d'un fichier .hyd (ou .hab avec les modèles LAMMI ou ASCII).

2021/11/26 11:46 · qroyer

2021/04/27 14:45 · qroyer

From:

<https://habby.wiki.inrae.fr/lib/tpl/bootstrap3-multilang/> - **HABBY**

Permanent link:

<https://habby.wiki.inrae.fr/lib/tpl/bootstrap3-multilang/doku.php?id=fr:develop>

Last update: **2021/04/27 14:39**

